A Watermark for Large Language Models

— Austin Houck, Tejasvi Yadav, Berk Gokmen, Arya Bhandari, Gemmechu Hassena —

github.com/austinhouck/deep-learning-final-project-sp25

1 Introduction

Large language models (LLMs) like ChatGPT and Gemini possess powerful text generation capabilities but also pose increasing risks of misuse across various applications. Detecting LLM-generated text is therefore essential for mitigating potential harms. The paper A Watermark for Large Language Models by Kirchenbauer and Geiping proposes a method for detecting such text through watermarking [1]. The watermarking technique works by selecting a randomized set of "green" tokens before each word is generated and then softly promoting the use of these tokens during sampling. The authors also introduce a statistical test to detect the presence of the watermark through z-scores. In our re-implementation of the paper, we were able to replicate similar results by testing the method on a subset of the data.

2 Chosen Results

The authors present three incremental watermarking methods that are easy to embed, cheap to detect, and robust for large language models. We aim to reproduce all three methods, focusing on two core results from the original paper. First, we replicate the case studies to determine under which conditions the watermark is detected, how it affects text quality, and how reliable it is. Second, we sweep the green-list size γ for the hard red-list watermark and both γ and the logit bias δ for the soft red-list watermark, to recreate the z-score versus sequence-length curves shown in Figure 3. This will demonstrate how different parameter settings influence watermark detectability. Together, these results highlight the fundamental challenge of balancing text quality against watermark robustness and show how parameter choices impact this trade-off.

Prompt	No Watermark (NW)	Watermark (W)	S	z	PPL (NW)	PPL (W)
...tled out of court and publicly reconciled. \nIn the '80s the band's popularity waned in the United States but remained strong abroad. Robin released three solo albums, with limited success. The Bee Gees	continued to tour, and Barry became a television producer. \nBut in the early '90s, the Bee Gees' popularity remained high. They scored a hit with "Don't Stop Believing" in 1990, and in 1992 the Bee Ge [...]	' 1990 album, "Spirits of the Century," was a mixed critical and commercial success.\nWhen the brothers were nominated for a Grammy Award in 1990, Mr. Gibb's "You Should Be Dancing" and "Massachusetts, [...]	0.68	12.73	3.15	1.93
...logged into their Google account and have verified profiles that match queries for the site. \nGoogle's John Mueller said there is no ranking benefit in using different Google Search Console and Google	Analytics data to calculate ranking sentiment. \nGoogle have a new feature called Keyword Difficulty Analysis that shows you the key words your competitors are ranking for. It shows the demand curve as [...]	+ accounts to see different numbers.\nGoogle also released their own great blog post on the news algorithm. They give lots of great advice to help your site do better.\nFinally, at the end of September [...]	0.77	11.88	9.24	4.10
...1 on the season with a 2.76 GAA and .906 save percentage. \nMar. 11 5:28 PM PT6:28 PM MT7:28 PM CT8:28 PM ETD:28 GMT8:28 S28 PM MST6:28 PM CST7:28 PM EST4:28 UAE (+1)20:28 ET21:28 BRT - Sergei Bobrovsky	stopped 28 of 30 shots in the Blue Jackets' 3-2 shootout loss to the Islanders on Sunday. The loss dropped him to 30-22-1 on the season with a 2.76 GAA and.906 save percentage. \nMar. 10 6:15 PM PT7:15 [...]	stopped 30 of 34 shots in the Blue Jackets' 6-2 win over the Canadiens on Monday. The victory pushed him to 31-21-1 on the season with a 2.72 GAA and.908 save percentage.\nMar. 10 11:49 AM PT12:49 PM [...]	0.62	2.40	1.33	1.45

Table 1: Comparison of outputs with and without watermarking, showing entropy (S), z-score (z), and perplexity (PPL) metrics from the paper [1].

3 Methodology

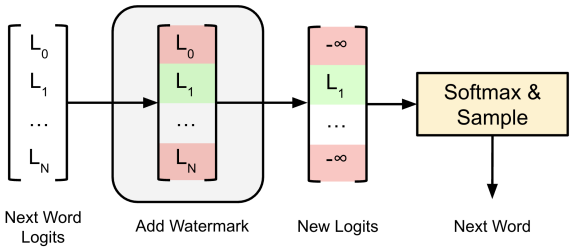

Figure 1: Hard Red List Watermarking Scheme

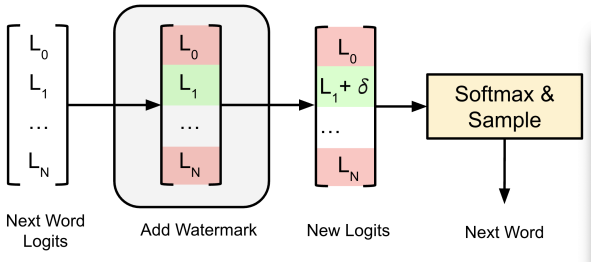

Figure 2: Soft Red List Watermarking Scheme

For our implementation of the paper's approach, we used a Jupyter Python notebook for all of our code and imported relevant libraries such as **pytorch** and **transformers**. Just like the paper, we relied on OPT-1.3B [2] as our language model and the RealNewsLike subset of the C4 dataset for most of our data. In the spirit of demonstrating how watermarking could help professors detect when students use LLMs to do their homework, we also included a few sample questions from the homework in our prompt dataset. To implement the watermarking schemes defined in the paper, including Hard Red List (Figure 1) and Soft Red List (Figure 2), we created two subclasses of **LogitsProcessor** from the **transformers** library, each tasked with modifying the logits output by the language model in accordance with their respective algorithms (i.e. setting red-list logits to negative infinity for algorithm 1, adding δ to green-list logits for algorithm 2, etc.). The ability to attach one of these watermarking mechanisms to an existing language model without the need for retraining is identified by the authors as a major advantage of this approach. Like the paper, we use z-score, spike entropy, and perplexity to compare the outputs of various prompts for both watermarking (W) and non-watermarking (NW) examples. By utilizing these metrics along with qualitative analysis of the watermarked outputs, we can determine the detectability of the watermarking schemes for various prompts along with the impact that watermarking has on text quality.

4 Results and Analysis

We ran our re-implementation of both the hard red-list and soft red-list watermarking on the three different case studies presented in the Chosen Results section, as well as across different parameters of γ and δ to create z-score vs. sequence length graphs. Due to limited compute resources and time, the graphs were created using 5 different prompts from the C4 dataset [3].

Table 2 includes the three different case studies. The first row is an example of a high-quality output, a watermark not apparent to the reader, that achieves a z-score of 8.68, which indicates the presence of a watermark since the z-score is greater than 4. Similar to the paper, the second case study showcases a watermark that fails to be detected due to the sequence length not being long enough. Finally, the last case study showcases a challenge we faced throughout our re-implementation. Due to unknown reasons (not mentioned in the paper), the LLM sometimes repeats itself over and over again for specific prompts. We believe that this is due to two factors: we used a small model (OPT-1.3B [2]), and the authors might have picked different parameters for the model to achieve the results presented in their table, such as a different value for temperature, seed, etc.

Prompt	No Watermark (NW)	Watermark (W)	S	z	PPL (NW)	PPL (W)
…While Arlington pored over the details, the Virginia General Assembly passed, and Democratic Gov. Ralph Northam signed, an incentives package worth up to $750 million for Amazon.	The package includes a $15 million cash grant, a 20 percent tax credit for Amazon's first five years of operations and $300 million in local property tax relief over the next decade…	Local officials said they were confident the deal would be approved.\nBut opponents of the deal have been waiting to hear the terms of the agreement…	0.84	8.68	8.47	8.48
…Services will include full security programs, compliance, third party vendor risk assessment, threat management, and managed security services.	\nKispert's appointment comes shortly after Grassi announced the retirement of Bruce Beyer, a 32-year veteran of the firm.	\n"I am excited to join the team at DMC and look forward to providing clients with the security and privacy services they need," Kispert said.	0.81	3.21	32.4	12.3
The vanishing gradient problem in Deep Learning is	a problem of finding a curve on a set of data points that represents the intensity of the point. The graph that the point should be in is the so called vanishing gradient.\n\nThe vanishing gradient problem is a very important one and it is one that has been studied extensively in the field of Deep Learning for decades…	…we are going to explore the application area of the vanishing gradient problem in deep learning by considering two different deep learning architectures, namely, the Linear Deep Learning Architecture and the Recurrent Deep Learning Architecture. We are also going to explore the application area of the vanishing gradient problem in deep learning by considering two different deep learning architectures, namely, the Linear Deep Learning Architecture and the Recurrent Deep Learning Architecture…	0.76	8.39	6.60	4.28

Table 2: Comparison of outputs with and without watermarking, showing entropy, Z-score, and perplexity (PPL) metrics.

Additionally, Figure 3 shows our results of the z-score vs. sequence length, which almost identically matches the paper. As expected, as sequence length increases, the ability to detect the watermark also increases. With a smaller green list γ, there is more restriction on word choice, so the z-score increases; similarly, with a higher logit bias δ, outputs become more predictable and the z-score increases as well.

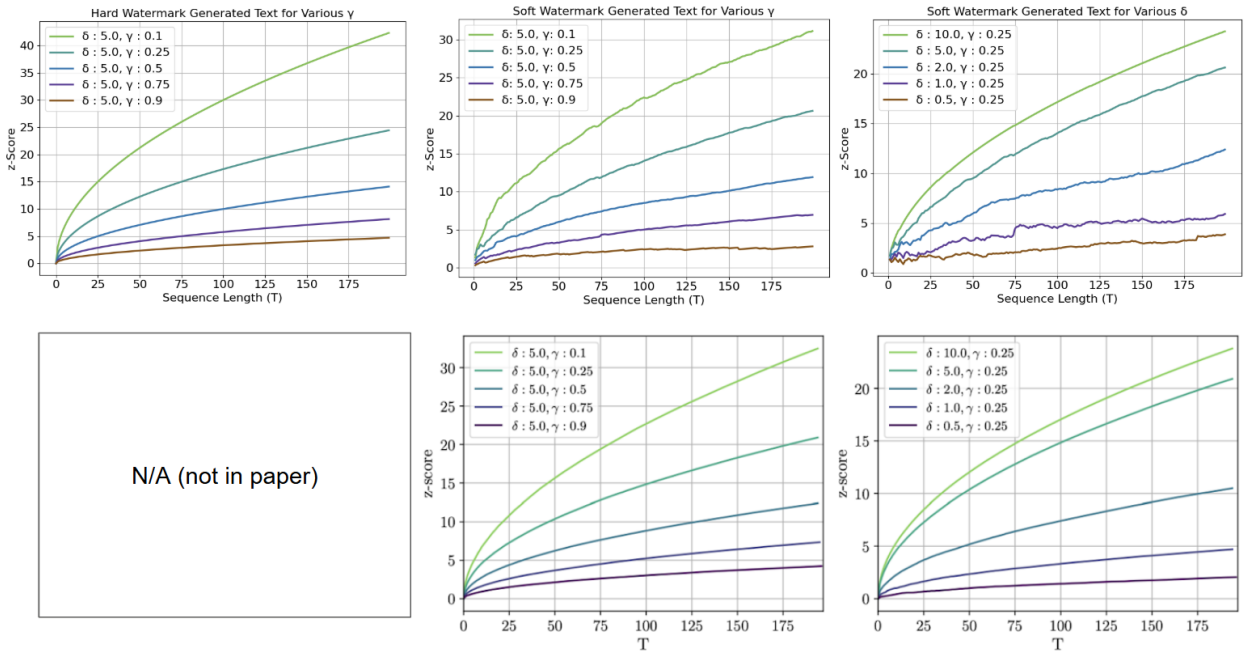

Figure 3: z-score vs. Sequence Length for original (bottom) and our re-implementation (top) of hard red-list and soft red-list watermarking for various values of γ and δ. The small noise is due to using a subset of the data.

In the context of the paper and the broader area of watermarking, our analysis shows that watermark detectability depends on sequence length and word choice. Priming the LLM to use specific words, either through restriction or by increasing logit scores, embeds a watermark in the output, but this comes at the cost of not allowing the LLM to choose the word it deems best, which results in worse text quality.

5 Reflection

During the course of re-implementing the paper's methodology, we found that the proposed algorithms indeed provided an efficient mechanism for the detection of LLM-generated text, though this sometimes comes at the cost of text quality. Algorithm 3 in particular provides a cryptographically secure approach that is robust to potential attacks. One opportunity for future work that we alluded to earlier is to use these watermarking schemes on larger, more robust LLMs, as the OPT-1.3B model is quite small compared to the current state-of-the-art. We would also like to see this tested on a broader variety of datasets to see how this might impact generation within certain specialized domains. In short, the authors provide a compelling case for the consideration of these watermarking techniques as a promising area of research in the broader context of AI risk mitigation.

References

[1] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, "A watermark for large language models," 2024. [Online]. Available: https://arxiv.org/abs/2301.10226

[2] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, "Opt: Open pre-trained transformer language models," 2022. [Online]. Available: https://arxiv.org/abs/2205.01068

[3] J. Dodge, M. Sap, A. Marasovic, W. Agnew, G. Ilharco, D. Groeneveld, and M. Gardner, "Documenting the english colossal clean crawled corpus," *CoRR*, vol. abs/2104.08758, 2021. [Online]. Available: https://arxiv.org/abs/2104.08758